NAME
> xclip − command line interface to X selections (clipboard)

SYNOPSIS
> **xclip** [OPTION] [FILE]...

DESCRIPTION
> Reads from standard input, or from one or more files, and makes the data available as an X selection for pasting into X applications. Prints current X selection to standard out.

> **−i**, **−in** Read text into X selection from standard input or files. (Default unless it would read from a tty.)

> **−o**, **−out**
>> Print the selection to standard out. (Default when no files are specified and standard input is a tty.)

> **−f**, **−filter**
>> Reads text into the X selection and causes xclip to print any text piped to standard input back to standard out unmodified. This does not work with files. Also requires xclip be invoked in the in mode with output level set to silent (the defaults if you are piping into xclip).

> **−selection** *selection*
>> Specify which X selection to use, options are "primary" (default), "secondary", or "clipboard".

> **−c**, **−clipboard**
>> Shorthand for "-selection clipboard".

> **−t**, **−target** *target*
>> Specify a particular data format using the given target atom (E.g., "text/html", "image/jpeg"). The default target is "UTF8_STRING". On output, if a selection doesn't provide "UTF8_STRING", xclip will fallback to "STRING".

> **−T** Show all valid targets for -t. Technically, this is just shorthand for "-target TARGETS". The special target atom named "TARGETS" contains the list of all valid target atoms for this selection. For more information about target atoms refer to ICCCM section 2.6.2.

> **−r**, **−rmlastnl**
>> Remove a newline at the end of the selection. Newline characters that are not the last character are not affected. If the selection does not end with a newline character, this option has no effect. This option is useful for copying one-line output of programs like **pwd** to the clipboard to paste it again into the command prompt without executing the line immediately due to the newline character **pwd** appends.

> **−l**, **−loops** *n*
>> Number of X selection requests (pastes into X applications) to wait for before exiting, with a value of 0 (default) causing xclip to wait for an unlimited number of requests until another application (possibly another invocation of xclip) takes ownership of the selection.

> **−d**, **−display** *display*
>> X display to use (e.g. "localhost:0"), xclip defaults to the value in $**DISPLAY** if this option is omitted.

> **−h**, **−help**
>> Show quick summary of options

> **−version**
>> Show version information.

> **−silent** Fork into the background to wait for requests, no informational output, errors only (default).

> **−quiet** Show informational messages on the terminal and run in the foreground.

> **−verbose**
>> Run in the foreground and provide a running commentary of what xclip is doing.

−noutf8

Operate in legacy (i.e. non UTF-8) mode for backwards compatibility. (Use this option only when really necessary, as the old behavior was broken).

xclip reads text from files or standard input and makes a copy available to other X applications for pasting as an X selection (traditionally with the middle mouse button). All files are read as input. When no files are specified, xclip checks whether standard input is a tty; if so, it switches to output mode and prints the contents of the selection to standard out. Otherwise, xclip presumes it is being piped into and reads from standard input. In short, xclip will do the right thing most of the time, but in complex cases or in carefully written shell scripts, you may want to force input or output modes with the **−i** or **−o** options.

xclip was designed to allow tighter integration of X applications and command line programs. The default action is to silently wait in the background for X selection requests (pastes) until another X application places data in the clipboard, at which point xclip exits silently. You can use the **−verbose** option to prevent xclip from backgrounding itself and see if and when it receives selection requests from other X applications.

X keeps track of three different selections: *primary*, *clipboard*, and *secondary*. In general, the primary selection is used to store short-lived data, such as when the user highlights text with a mouse. The clipboard stores longer lived data and only changes when the user explicitly requests it by pressing cut or copy. Selections can store any sort of data, but usually one expects to see non-text datatypes, such as images, in the clipboard. The secondary selection is rarely used by programs, but some users find it handy as a quick way to stash a copy of the primary selection that isn't quite as ephemeral.

A selection can have multiple "targets", each of which is a different datatype. By default xclip uses the target "UTF8_STRING". For example, if you copy text from a web browser, it may have the choice to paste it as either "UTF8_STRING" or "text/html". You can see what targets are available using the **-T** option and you can pick a specific one using **-t**.

Options can be abbreviated as long as they remain unambiguous. For example, it is possible to use **−d** or **−disp** instead of **−display**. However, **−v** couldn't be used because it is ambiguous (it could be short for **−verbose** or **−version**), so it would be interpreted as a filename.

Note that only the first character of the selection specified with the **−selection** option is important. This means that "p", "sec" and "clip" would have the same effect as using "primary", "secondary" or "clipboard" respectively.

EXAMPLES

I hate man pages without examples!

uptime | xclip

Put your uptime in the X selection. Then middle click in an X application to paste.

xclip -loops 10 -verbose /etc/motd

Exit after /etc/motd (message of the day) has been pasted 10 times. Show how many selection requests (pastes) have been processed.

xclip > helloworld.c

Put the contents of the selection into a file.

xclip -t text/html index.html

>   Copy a file with a specific MIME type. Middle click in an X application supporting HTML to
>   paste the contents of the given file as HTML.

xclip -c -T

>   List valid data formats available on the clipboard. For example, after doing right-click "Copy Im-
>   age" in a web browser, one might see:

```
TIMESTAMP
TARGETS
MULTIPLE
SAVE_TARGETS
image/png
image/tiff
image/jpeg
```

xclip -c -t image/jpeg < foo.jpg

>   Copy an image to the clipboard.

xclip -c -t image/jpeg > bar.jpg

>   Paste an image from the clipboard to a file.

ENVIRONMENT
DISPLAY

>   X display to use if none is specified with the **--display** option.

SEE ALSO

>   *xclip-copyfile*(1), *xclip-cutfile*(1), and *xclip-pastefile*(1) copy and move files via the X clipboard.
>
>   *xsel*(1) and *wl-clipboard*(1) are command line programs similar to xclip. xsel has the ability to keep a pri-
>   mary selection even after a program has closed. wl-clipboard works with Wayland instead of X.
>
>   *copyq*(1), *diodon*(1), *gpaste*(1), and *xclipboard*(1) are just a few of many, many GUI clipboard managers
>   which let you browse your clipboard's history.

KNOWN BUGS

>   xclip is not ICCCM compliant. For example, the TIMESTAMP isn't set.
>
>   Only one target type can be chosen. Would it be useful to let users pick a different type for each file?

REPORTING BUGS

>   Please report any bugs, problems, queries, experiences, etc. via github. ⟨https://github.com/astrand/xclip⟩

AUTHORS

>   Peter Åstrand ⟨astrand@lysator.liu.se⟩
>   Kim Saunders ⟨kims@debian.org⟩